

IN THE CLAIMS

Please amend the claims as follows.

- 1 1. (Previously Presented) An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 an object oriented program residing in the memory comprising a plurality of
5 instructions; and
6 a dynamic compiler residing in the memory and executed by the at least one
7 processor, the dynamic compiler being invoked during execution of the object oriented
8 program, the dynamic compiler allocating at least one object in the object oriented
9 program to an invocation stack frame for a method that allocates the at least one object,
10 the dynamic compiler comprising:
11 an escape analysis mechanism that marks each instruction that
12 allocates a new object as one of global escape, no escape, and arg escape
13 based on information available from previously-loaded classes that are part
14 of the object oriented program; and
15 an object allocation mechanism that allocates at least one object
16 that is created by an instruction marked as no escape by the escape analysis
17 mechanism to an invocation stack frame for a method that allocates the
18 object;
19 wherein the dynamic compiler analyzes each class as it is loaded to determine
20 whether the newly-loaded class affects the allocation of an object by the object allocation
21 mechanism to the invocation stack frame, and if so, the dynamic compiler changes the
22 allocation of the object to a heap.

1 2. (Cancelled)

1 3. (Cancelled)

1 4. (Previously Presented) An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 an object oriented program residing in the memory comprising a plurality of
5 instructions; and
6 a dynamic compiler residing in the memory and executed by the at least one
7 processor, the dynamic compiler being invoked during execution of the object oriented
8 program, the dynamic compiler allocating at least one object in the object oriented
9 program to an invocation stack frame for a method that allocates the at least one object,
10 wherein the dynamic compiler changes the allocation of the object from the
11 invocation stack frame to a heap due to information that becomes available from at least
12 one class that is loaded after the dynamic compiler allocates the at least one object to the
13 invocation stack frame.

1 5. (Original) The apparatus of claim 4 wherein the dynamic compiler changes at least
2 one pointer to the object allocated on the invocation stack to point to an object allocated
3 on the heap as a result of information that becomes available as more classes that are part
4 of the object oriented program are loaded.

1 6. (Previously Presented) An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 an object oriented program residing in the memory comprising a plurality of
5 instructions;
6 a portion of the object oriented program that is selected for dynamic compilation;
7 a dynamic compiler residing in the memory and executed by the at least one
8 processor, the dynamic compiler being invoked during execution of the object oriented
9 program, the dynamic compiler comprising:
10 an escape analysis mechanism that marks each instruction that allocates a
11 new object as one of global escape, no escape, and arg escape based on
12 information available from classes that are part of the object oriented program that
13 have been previously loaded at run-time; and
14 an object allocation mechanism that allocates at least one object that is
15 created by an instruction marked as no escape by the escape analysis mechanism
16 to an invocation stack frame for a method that allocates the object;
17 wherein the dynamic compiler changes the allocation of the object from the
18 invocation stack frame to a heap due to information that becomes available from at least
19 one class that is loaded after the dynamic compiler allocates the at least one object to the
20 invocation stack frame.

1 7. (Previously Presented) A method for allocating objects to memory in an object
2 oriented program during dynamic compilation of a portion of the object oriented program
3 while the object oriented program is executing, the method comprising:
4 (A) determining whether compilation of the portion is needed;
5 (B) if compilation of the portion is needed:
6 (B1) analyzing each instruction in the portion that allocates a new object;
7 (B2) allocating at least one object that is created by an instruction to an invocation
8 stack frame for a method that allocates the at least one object;
9 wherein acts (B1) and (B2) comprise:
10 marking each instruction in the portion that allocates a new object as one
11 of global escape, no escape, and arg escape based on information available from classes
12 that are part of the object oriented program that have been previously loaded at run-time;
13 and
14 allocating at least one object that is created by an instruction marked as no
15 escape by the escape analysis mechanism to an invocation stack frame for a method that
16 allocates the at least one object, and
17 wherein the dynamic compiler analyzes each class as it is loaded to determine
18 whether the newly-loaded class affects the allocation of an object to the invocation stack
19 frame, and if so, changing the allocation of the object to a heap.

1 8. (Original) The method of claim 7 wherein act (A) comprises determining whether a
2 method in the portion has been executed a number of times equal to or greater than a
3 predetermined threshold value.

1 9. (Cancelled)

1 10. (Previously Presented) A method for allocating objects in an object oriented program
2 to memory, the method comprising:
3 loading a plurality of classes that are part of the object oriented program;
4 executing code from at least one of the plurality of loaded classes;
5 determining whether dynamic compilation of a portion of the object oriented
6 program is needed;
7 if dynamic compilation of the portion is needed, allocating at least one object to
8 an invocation stack frame for a method that allocates the at least one object;
9 analyzing compiled code as each subsequent class in the object oriented program
10 is loaded; and
11 changing the allocation of the at least one object from the invocation stack frame
12 to a heap.

1 11. (Cancelled).

1 12. (Cancelled)

1 13. (Currently Amended) A program product comprising:
2 a dynamic compiler that is invoked during execution of an object oriented
3 program, the dynamic compiler allocating at least one object in the object oriented
4 program to an invocation stack frame for a method that allocates the at least one object,
5 wherein the dynamic compiler comprises:
6 an escape analysis mechanism that marks each instruction that allocates a
7 new object as one of global escape, no escape, and arg escape based on information
8 available from previously-loaded classes that are part of the object oriented program; and
9 an object allocation mechanism that allocates at least one object that is
10 created by an instruction marked as no escape by the escape analysis mechanism to an
11 invocation stack frame for a method that allocates the at least one object;
12 wherein the dynamic compiler analyzes each class as it is loaded to
13 determine whether the newly-loaded class affects the allocation of an object by the object
14 allocation mechanism to the invocation stack frame, and if so, the dynamic compiler
15 changes the allocation of the object to a heap; and
16 tangible signal bearing media bearing the dynamic compiler.

1 14. (Original) The program product of claim 13 wherein the signal bearing media
2 comprises recordable media.

1 15. (Original) The program product of claim 13 wherein the signal bearing media
2 comprises transmission media.

1 16. (Cancelled)

1 17. (Cancelled)

1 18. (Currently Amended) A program product comprising:
2 (A) a dynamic compiler that compiles a portion of an object oriented program, the
3 dynamic compiler being invoked during the execution of the object oriented program, the
4 dynamic compiler comprising:
5 (A1) an escape analysis mechanism that marks each instruction that
6 allocates a new object as one of global escape, no escape, and arg escape based on
7 information available from classes that are part of the object oriented program that
8 have been previously loaded at run-time;
9 (A2) an object allocation mechanism that allocates at least one object that
10 is created by an instruction marked as no escape by the escape analysis
11 mechanism to an invocation stack frame for a method that allocates the object,
12 wherein the dynamic compiler changes the allocation of the object from
13 the invocation stack frame to a heap due to information that becomes available from at
14 least one class that is loaded after the dynamic compiler allocates the at least one object to
15 the invocation stack frame; and
16 (B) tangible signal bearing media bearing the dynamic compiler.

1 19. (Original) The program product of claim 18 wherein said signal bearing media
2 comprises recordable media.

1 20. (Original) The program product of claim 18 wherein said signal bearing media
2 comprises transmission media.

1 21. (Currently Amended) A program product comprising:

2 (A) a dynamic compiler that compiles a portion of an object oriented program, the
3 dynamic compiler being invoked during the execution of the object oriented program and
4 allocating at least one object in the object oriented program to an invocation stack frame
5 for a method that allocates the at least one object, wherein the dynamic compiler changes
6 the allocation of the object from the invocation stack frame to a heap due to information
7 that becomes available from at least one class that is loaded after the dynamic compiler
8 allocates the at least one object to the invocation stack frame; and

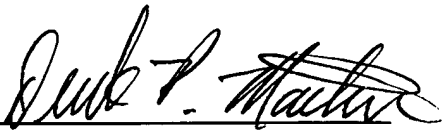
9 (B) tangible signal bearing media bearing the dynamic compiler.

1 22. (Previously Presented) The program product of claim 21 wherein said signal bearing
2 media comprises recordable media.

1 23. (Previously Presented) The program product of claim 21 wherein said signal bearing
2 media comprises transmission media.

The claims section is hereby re-submitted in response to the Notice of Non-Compliant Amendment dated 10/19/05. Applicant respectfully requests entry of the amendment and allowance of the pending claims.

Respectfully submitted,

By 
Derek P. Martin
Reg. No. 36,595

MARTIN & ASSOCIATES, L.L.C.
P.O. Box 548
Carthage, MO 64836-0548
(417) 358-4700